AISRP Grant NNG06GE95G
Annual Progress Report

Year II: 2007-08

Project Title: **Adaptive Algorithms for Optimal Classification and Compression of Hyperspectral Images**

PI: Prof. Y.Q. Chen
(New) Electrical & Computer Engineering
Utah State University
4120 Old Main Hill
Logan, UT 84321-4120

Co-I: Prof. Tamal Bose
(Original PI) Virginia Tech
Bradley Dept. of Electrical and Computer Engineering
435 Durham, MS 350
Blacksburg, VA 24061

Co-I: Prof. Erzsébet Merényi
Electrical & Computer Engineering
Rice University
6100 Main Street MS 380
Houston, TX 77005

Date: February 5, 2008

Brief Summary of Research Project

The goal of this project is to design and implement efficient algorithms for the real-time compression of hyperspectral images. The uniqueness of this project is that the compression algorithms will be recursively optimized as a function of the performance of classification (or unsupervised clustering) algorithms on the same data. Certain performance metrics will be defined for the classification schemes and some others for the compression algorithms. A combination of these metrics will be used to design a cost function, which in turn will be optimized to update the compression and restoration algorithms. In other words, the performance of the classification algorithms will dictate the real-time adaptation of the digital filters used in the compression schemes. In this new concept, the filters in the compression algorithms, instead of acting independently, form a coupled system with the classification (or clustering) algorithms. This coupled system will be applied to the spectral dimension in this project since compression along the spectral axis is much less researched and understood than compression in the spatial

domain. If it works as expected, we can also explore a combined spatio-spectral application.

Using an iterative neural learning algorithm, called the Generalized Relevance Learning Vector Quantization, the classification and feature extraction are welded together with a 2-way feedback loop, and the classification cannot be separated and substituted by a different classifier. In our proposed work, the classification algorithms can be replaced by others if so desired.

The classifications are performed by a neural (ANN) classifier developed by Co-I EM. The reason for this is that hyperspectral imagery (100-500 input variables) containing as many classes as we include in this study have not been classified with better accuracy than this ANN classifier is capable of. Therefore, other classifiers are not as good for measuring the classification accuracy at different compression ratios as this tool. In real on-board compression scenarios the ability to use of a large number of (meaningful) classes is important for the representation of an application domain, therefore we need to simulate this aspect in our optimization cycle.

Research Progress in Year-II

1. (Rice) Unsupervised clustering: We did benchmark clustering on our uncompressed hyperspectral test image, and are performing the same on compressed-decompressed images. Currently the data mining and similar fields do not offer a suitable measure for the quality of clustering for a) high-dimensional data b) data with many clusters of widely varying shapes, sizes, densities, and overlaps. We developed (mostly under a parallel AISRP grant) a new cluster validity metric in order to be able to compare the quality of these clustering in a quantitative way.

2. (VT) Developed the mathematical algorithms for optimizing compression algorithms: (a) adaptive filters, (b) predictor models, (c) adaptive quantizers, and (d) transform coders. This effort required constructing appropriate mathematical cost functions (from the insight gained in Year 1) and we derived several different optimizing approaches.

3. (VT) Implemented Scheme 1 (Figure 5 in proposal). The compression scheme used is an Adaptive Differential Pulse Code Modulation (ADPCM) transmitter, which is composed of an adaptive predictive filter, adaptive quantizer, and an algorithm to update the filter coefficients in real time. The compressor is followed by a decompressor, which is an ADPCM receiver. The receiver consists of the same predictive filter and the same algorithm as the transmitter. The transmitter and receiver usually run in synchronization. The decompressed image has a certain Signal to Noise Ratio (SNR) depending upon the desired compression ratio and the performance of the adaptive filter and quantizer. A low SNR typically has an adverse effect on the classification algorithm. The performance metrics after classification are compared to a desired set of metrics. The error is used to compute a cost function that is minimized by an algorithm to update the coefficients of the predictive filter. We

have used some simple cost functions. These cost functions are minimized an adaptive filtering algorithm such as the LMS, EDS or RLS.

4. (Rice) Supervised classification of compressed and decompressed images yielded excellent results: Classification accuracy of these images remains within 2% of the classification accuracy on original uncompressed data (Table 1). Some representative classes of spectral signatures are given in Appendix A. We continue the optimization approach with feedback from classification quality, and aim to achieve as good or better classification accuracy on compressed data as on the original data. Because we simulate ANNs on sequential computers of medium capacity (Sun Blade machines) the production of classifications, including an initial overhead of setting, verifying and testing the learning process itself, is fairly long, and also dependent on the demand on shared resources such as network and disk space.

5. (USU) Provided project administration, accounting, and oversight.

Table 1: Measuring the effect of compression on classification accuracy. Data: Hyperspectral image of Lunar Crater Volcanic Field, 196 spectral bands, 614 x 420 pixels. Classifications were done for 23 known surface cover types. Original uncompressed data are labeled with "LCVF", a compressed-uncompressed data set with "D1c16".

Data Set	Run C	Run A	Run E	Run M	Run B.0		Avg.	Std.
D1c16	84.9	86.3	85.1	82.4	84.6		84.66	1.46
	Run 1	Run 2	Run 5.1	Run 5.2	Run 6			
LCVF benchmark	86.01	86.03	86.05	86.15	86.1		86.07	0.06

Publications

The following publications have resulted from this project (or in part).

1. M. Radenkovic and Tamal Bose, "A recursive blind adaptive identification algorithm and its almost sure convergence," **IEEE** *Trans. Circuits and Systems, Part-I*, pp. 1380-1388, June 2007.

2. Tamal Bose, M.-Q. Chen, and R. Thamvichai, "Stability of the 2-D Givone-Roesser model with periodic coefficients," *IEEE Trans. Circuits and Systems, Part-I,* pp. 566-578, Mar. 2007.

3. B. Xie, Tamal Bose and E. Merenyi, "Novel algorithms for optimal compression using classification metrics," *IEEE Aerospace Conference***,** March 2008 (accepted and in press).

4. B. Xie, Tamal Bose, and E. Merenyi, "New Algorithms for the Classification and Compression of Hyperspectral Images," *Proc. of NASA Science and Technology Conference,* May 2007.

5. Erzsébet Merényi, William H. Farrand, James V. Taranik, and Timothy B. Minor (2007) Classification of Hyperspectral Imagery with Neural Networks: Comparison to Conventional Tools. Under review for the May, 2008 Special Issue of Photogrammetric Engineering & Remote Sensing, Submitted July, 2007.

6. Tasdemir, K, and Merényi, E. (2007) Exploiting the Data Topology in Visualizing and Clustering of Self-Organizing Maps. Submitted to *IEEE* Trans. *Neural Networks*.

7. Taşdemir, K. and Merényi, E. (2007) A new cluster validity index for prototype based clustering algorithms based on inter- and intra-cluster density. *Proc. Int'l Joint Conf. on Neural Networks (IJCNN 2007),* Orlando, FL, August 12 – 17, 2007. IEEE Catalog number 07CH37922C. (Journal version in preparation.)

Research Plan for Year-III

1. (VT) Implement Scheme 2 (Figure 6 in proposal) in conjunction with the developed algorithms in Year II. Transfer the transform and quantized domain images to Rice.

2. (Rice and VT) Perform classification of hyperspectral images in the transform and quantized domain obtained from 1. This will include experiments with a variety of transforms, compression ratios and quantizers. Send classification metrics and data back to USU for integration into the algorithms of Scheme 2 and modifying the algorithms, as necessary. The adaptive algorithms for the transform coder and the quantizer have to be finalized based on several iterations of information exchange between USU and Rice.

3. (Rice and VT) Intermediate variants: As mentioned in the last paragraph of the description of Scheme 2, there are two extreme cases of the availability of training data for classification. One, where there is sufficient data is available, and the other when we have no labeled training data. There are many scenarios in between where some knowledge is available on classification. In this task, we will implement and evaluate our algorithms for a representative number of these cases and accordingly modify our algorithms, as described in "Proposed Methods," after Scheme 2.

4. (VT, USU, Rice) Create demo of all our algorithms and project web site. All of our compression algorithms, codes, data analysis, and images will be available on our project web site for interested users.

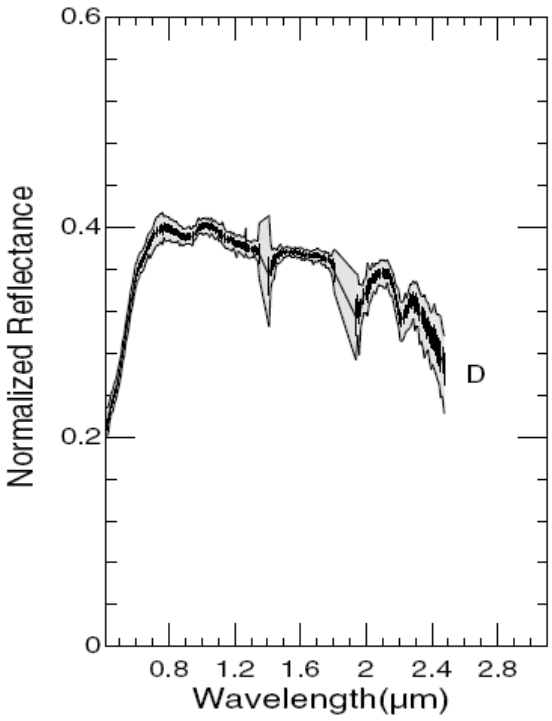
Most of our work during the last year appears in an impending publication, which is given in Appendix B. This publication contains many of the interesting results and applications, which illustrate the performance of the developed algorithms with realistic data.

APPENDIX A

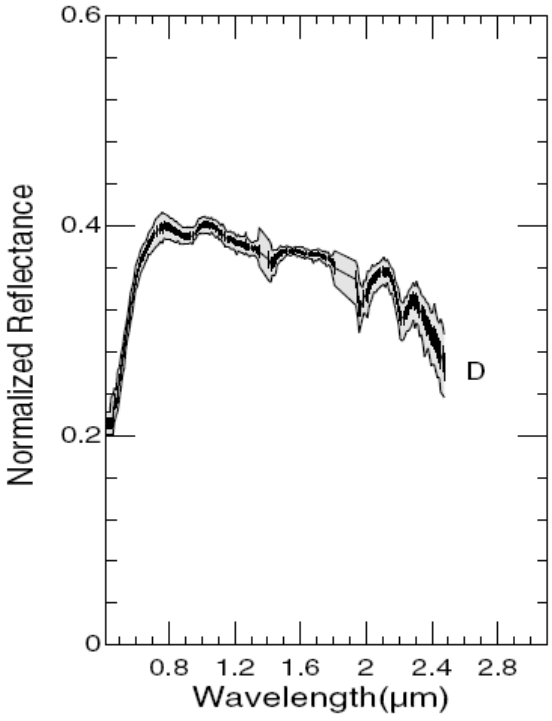
Representative spectral signal comparisons

Spectral signature comparison (Mean, STD, Envelope)

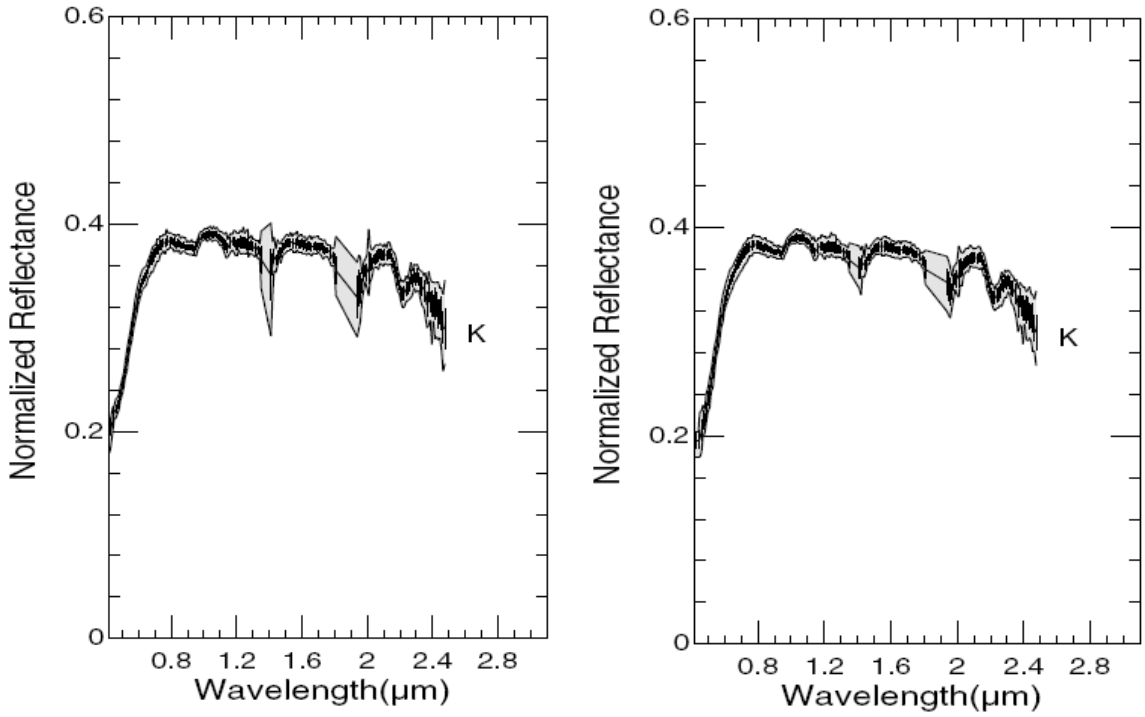
LCVF Uncompressed Data



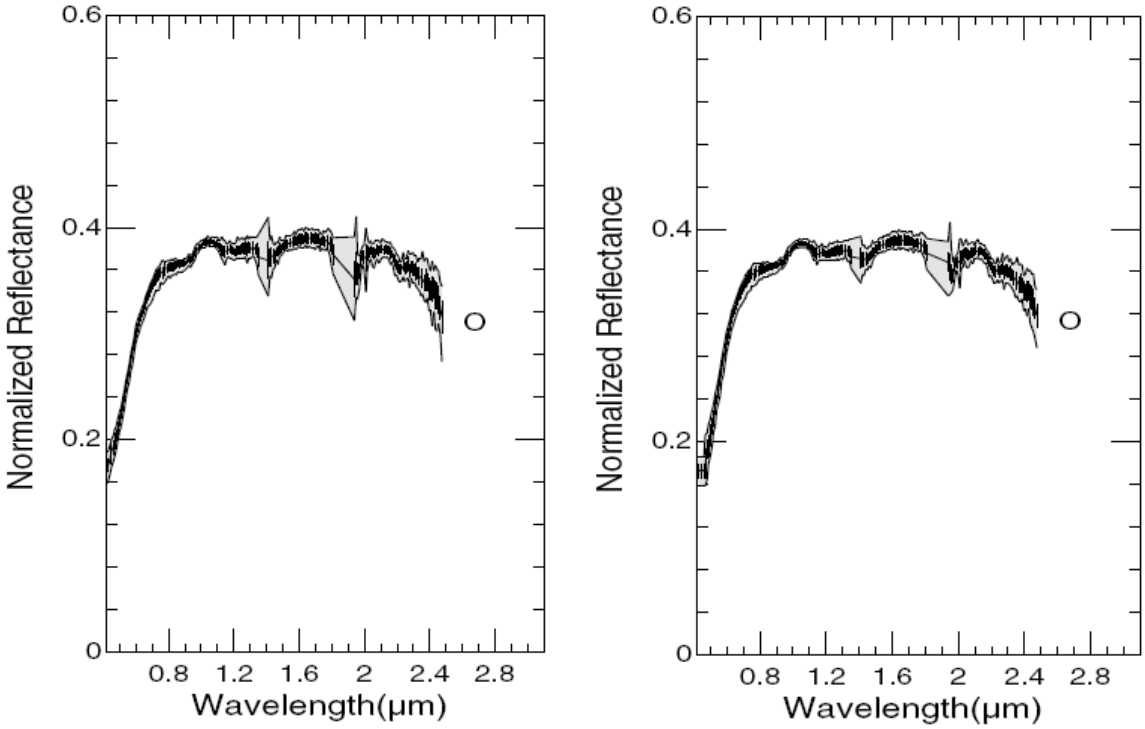
LCVF Compressed and Decompressed Data



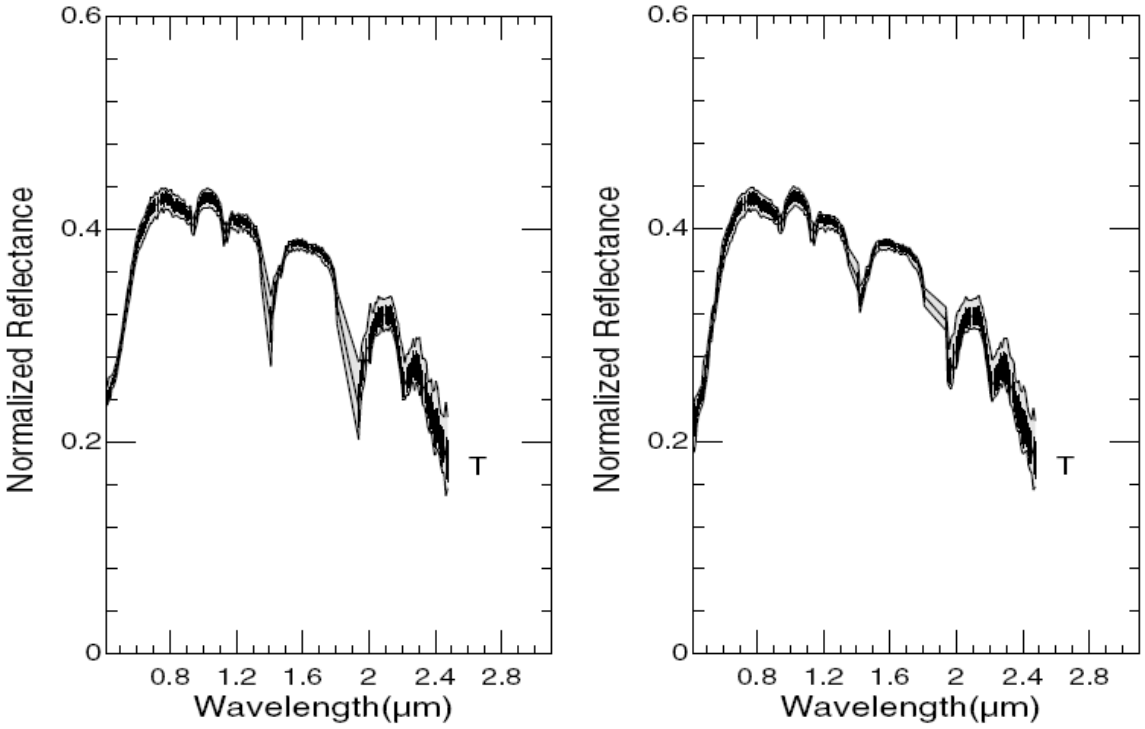
Class D



Class K



Class O


0.6
0.4
0.2
0
Normalized Reflectance
T
0.8 1.2 1.6 2 2.4 2.8
Wavelength(μm)
0.6
0.4
0.2
0
Normalized Reflectance
T
0.8 1.2 1.6 2 2.4 2.8
Wavelength(μm)

Class T

APPENDIX B

(Paper to appear in the IEEE Aerospace Conference, March 2008)

This paper contains many of the results and applications performed during Year-2 of this grant.

Novel Algorithms for Optimal Compression Using Classification Metrics

Bei Xie and Tamal Bose
Wireless @ Virginia Tech
Bradley Department of Electrical and Computer Engineering
Virginia Tech
Blacksburg, VA 24061
540-231-2964
beixie@vt.edu, tbose@vt.edu

Erzsébet Merényi
Electrical and Computer Engineering
Rice University
6100 Main Street MS 380
Houston, Texas 77005
713-348-3595
erzsebet@rice.edu

Abstract—In image processing, classification and compression are very common operations. Compression and classification algorithms are conventionally independent of each other and performed sequentially. However, some class distinctions may be lost after a minimum distortion compression. In this paper, two new schemes are developed that combine the compression and classification operations in order to optimize some classification metrics. In other words, the compression systems are improved under classification constraints. In the first scheme, compression is achieved by using Adaptive Differential Pulse Code Modulation (ADPCM). Optimization of filter coefficients is done by using a simple genetic algorithm (GA). In the second scheme, compression is achieved by image transform and quantization. The parameters in transform and quantization are adapted to improve the compression system and reduce the classification errors. Computer simulations are performed on hyperspectral images. The results are promising and illustrate the performance of the algorithms under various classification constraints and compression schemes. [1] [2]

TABLE OF CONTENTS

1. INTRODUCTION

Image classification is used to find special features of an image. It is usually performed on the original image, not the image after compression and decompression processing. Because of the large storage volume and spectral and spatial sample data redundancy [1], it is economical to compress hyperspectral images before classification. Obviously, there can be classification error between pre- and post-compression classifications if the compression is lossy. Furthermore, some class distinctions may be lost after a minimum distortion compression. To reduce classification error, it is necessary to improve the compression system with feedback from the classifier.

In this paper, we used clustering to map image data into baseline clusters, and this approach serves the role of classification for this study. The compression and the clustering systems are combined as in Figure 1. The basic idea of this system is to implement clustering on both the original uncompressed image and the image after compression and decompression operations. Clustering is done by using a simple unsupervised Self Organizing Map (SOM). The clustering error between pre- and post-compression is fed back to improve the compression system. In this process, the quality of clustering will gradually increase, and the compression scheme will gradually improve in order to capture the important clustering features.

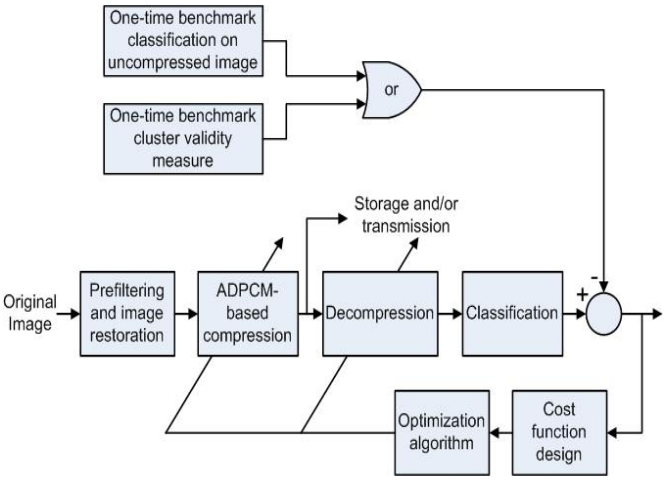


Figure 1 – System combining compression and clustering

With this system, two schemes are developed. In the first scheme, compression is achieved by using ADPCM, which has an adaptive predictor filter. The predictor coefficients are updated in real-time by optimizing a cost function based on clustering errors. A GA is used to choose the best filter coefficients in ADPCM according to the cost function. A corresponding algorithm, named GA-ADPCM algorithm, is developed. In the second scheme, compression is achieved

[1] 1-4244-1488-1/08/$25.00 ©2008 IEEE. This work was supported in part by NASA Grant # NNG06GE95G.
[2] IEEEAC paper #1080, Version 3, Updated Dec 12, 2007

by image transform and quantization. The parameters in transform and quantization are updated according to the clustering error. An algorithm, named Feedback-transform, is developed to realize this scheme.

In our experiments, a SOM clustering is used, since SOMs have been successfully used to extract useful information from a number of space mission data sets, as well as to perform terrestrial studies [2]. For the ADPCM scheme, two ADPCM predictors are tested to minimize the mean-square error in the compression system. One is a Least Mean Square (LMS) predictor, and the other is a Euclidian Direction Search (EDS) predictor. LMS is a very popular adaptive filter algorithm. EDS, developed by Xu and Bose [3], [4], is a relatively new algorithm that aims to increase the convergence speed of the LMS. A GA is used to minimize the clustering error since GA provides a general approach for searching for global minima or maxima within a bounded, quantized search space [5]. For the transform scheme, a popular transform coding Discrete Cosine Transform (DCT) is tested.

2. ADPCM SCHEME

In this scheme, the compression block uses ADPCM algorithms. A GA is used to optimize the filter coefficients of ADPCM, which tries to achieve minimal clustering error. ADPCM algorithms try to minimize the mean-square or least square error. Therefore, the ADPCM scheme aims to achieve the minimum of a signal error metric and clustering error. The corresponding algorithm, GA-ADPCM, is developed.

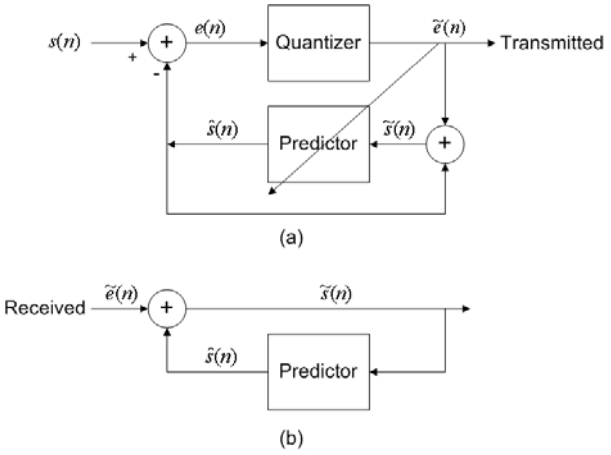


Figure 2 – ADPCM: (a) transmitter, (b) receiver

Review of ADPCM System

Figure 2(a) and (b) [4] show the transmitter and receiver of the ADPCM system, respectively. The signal to be transmitted or stored is $s(n)$. The output of the adaptive predictor is $\hat{s}(n)$. The input to the adaptive predictor is $\tilde{s}(n)$. The difference between the actual data and the predicted data is $e(n) = s(n) - \hat{s}(n)$. The quantized error to be transmitted or stored is $\tilde{e}(n)$. In this paper, an adaptive quantizer named Jayant quantizer [4] is used. Two adaptive filter algorithms, LMS and EDS, are used. The cost function takes the mean square of the error and thereby is quadratic.

Review Genetic Algorithm

A genetic algorithm is used to find approximate solutions to optimization and search problems using techniques inspired by evolutionary biology [6]. The process is shown in Figure 3 [7]. This process is repeated until the final goal is achieved. In our paper, the individuals are the coefficients in the ADPCM filter. The fitness criterion is a cost function based on the clustering error between pre- and post-compression images.

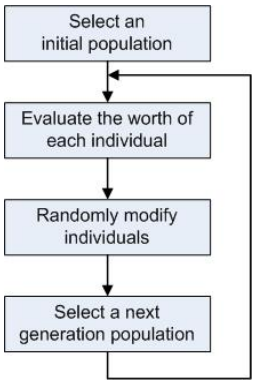


Figure 3 – Genetic algorithm

GA-ADPCM Algorithm

Since the genetic algorithm (GA) has the property of finding the optimal solution and the ADPCM algorithm has the property of minimizing an error metric, these two algorithms are combined into the GA-ADPCM algorithm that can not only minimize the mean-square error but also minimize the clustering error between pre- and post-compression images. The flowchart is shown in Figure 4. First, an initial population of random individuals is generated, which are the different sets of coefficients used in the ADPCM predictor. Second, the ADPCM compression and decompression processing is implemented using these different sets of coefficients. Third, the SOM is applied to the decompressed images. The application of SOM to the original image should be done at the beginning. Fourth, the clustering error between the decompressed image and the original image is calculated. The clustering errors of all individuals are sorted. Next, 50% of the fittest individuals are chosen and applied genetic operations, e.g., crossover and mutation. Finally, the new coefficients population is created and the procedure is repeated again.

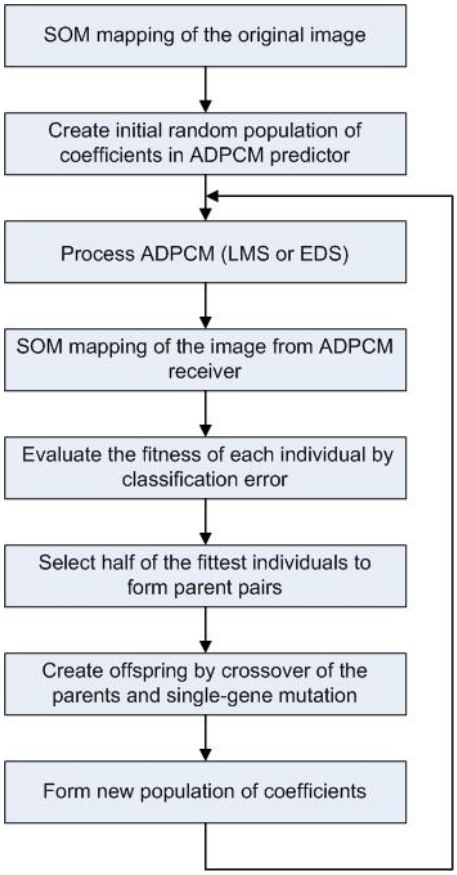


Figure 4 – Genetic ADPCM algorithm

Cost Function

The fitness function F to be minimized in GA is defined as:

$$F = C_e / N \quad (1)$$

where C_e is the number of pixels clustered incorrect, and N is the total pixels of the image. F becomes the percentage of incorrectly classified mapped pixels. C_e is obtained by the following steps: First, we do $C_m = C_o - C_g$, where C_o is the matrix containing the clustering result of the original image, which has all pixels assigned to different clusters. C_g is the matrix containing the clustering result of the image after ADPCM compression and decompression system. C_m is the difference between the two clustered images. Second, we assign all the nonzero points in the C_m matrix to be 1 and add them together to get the clustering error C_e.

The cost function of this algorithm is no longer quadratic, but a more complex function. For instance, it may appear as in Figure 5. In each part, A, B, and C, respectively, the cost function is quadratic because of the properties of the ADPCM. The LMS and EDS predictors try to find one of the local minima of the cost function, which means trying to find one of the points A, B, or C. However, they may not guarantee the global minimum. Since GA is then used to find the approximate global minimum of the whole cost function, the algorithm GA-ADPCM tries to find point A in this example.

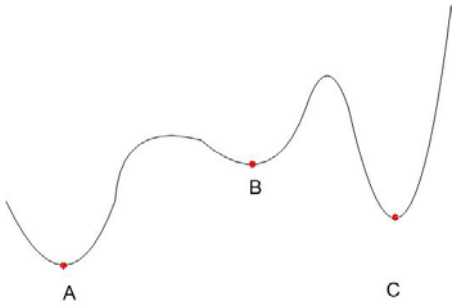


Figure 5 – Cost function of GA-ADPCM

3. TRANSFORM SCHEME

In this scheme, the compression block uses a transform algorithm and a quantization. The transform concentrates the energy of the image to a small number of pixels [4]. These pixels can be quantized with more bits than other pixels. In this paper, the DCT is used because it is a popular transform algorithm. The DCT has several advantages, such as the coefficients are real, it has near-optimal property for energy compaction, and it is computationally efficient [4].

Hyperspectral images are high data volume images. For DCT processing, a hyperspectral image needs to be divided into blocks. Suppose a full length DCT is used. As the property of DCT, the energy is always located in the low frequency pixels. It needs to use larger bits to quantize these significant pixels. Usually, the greater the block size, the less total DCT processing time is needed. However, it needs more bits for the significant pixels to achieve a certain clustering error, which increases the total storage. Obviously, for a fixed number of quantization bits, using the shorter length of block size will get better results, but a longer processing time is needed. Depending on the different limitations on processing speed or storage capacity, one can either adjust the block size or the number of bits. Furthermore, one can adjust both of the parameters, and find a tradeoff between the length of block size and the number of bits. For two different limitations, two methods are provided. Method I, given a fixed number of bits, find the largest block size. Method II, given a fixed block size, find out the smallest number of bits needed. Of course, the prerequisite of these two methods is to keep a certain clustering correction rate. The Feedback-transform algorithm is developed to realize these two methods. One can also use this algorithm to find a balance between the length of block size and the number of bits.

Feedback-Transform Algorithm

The diagram of the feedback transform algorithm is shown in Figure 6. At the beginning, initial parameters are set up. One of two quantization methods, Method I and Method II, can be chosen in this scheme. For Method I, a large block size is used. For Method II, a small number of bits is used

for quantization. After applying transform and clustering, the result is compared with the clustering result of the original image. The fitness function is still calculated by using equation (1). If the clustering error does not meet the requirement, then the parameters are adjusted. For Method I, it means to cut the block size into half. For Method II, it means to increase the number of bits. Then these steps are repeated until the clustering error threshold is achieved.

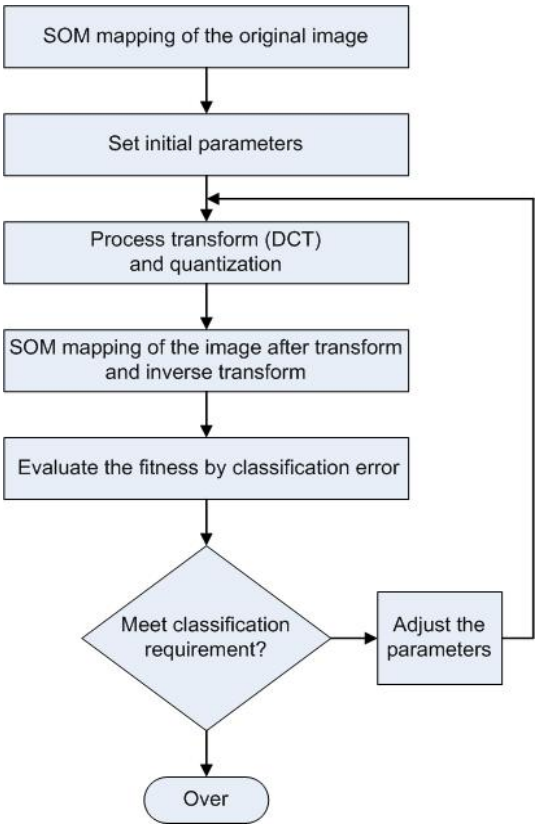


Figure 6 – Feedback-transform algorithm

4. SIMULATIONS

An example of a hyperspectral cube is shown in Figure 7. This is the Lunar Crater Volcanic Field (LCVF) in Nevada, one of NASA's standard remote sensing imaging sites. In this experiment, two hyperspectral cubes were used. One is the LCVF, the other is of the Jasper Ridge (JR) site. Both were acquired by the AVIRIS hyperspectral sensor [8].

GA-ADPCM Performance

One image band of the Jasper Ridge was used. The subimage with 512x512 pixels was used to be the original image in the simulation and shown in Figure 8. To save the storage of the memory on the computer and increase the speed of performance, the image was divided into blocks. Different block sizes were tried, including 16x16, 32x32, and 64x64 pixels. In SOM, different numbers of clusters were also tried, including 3 clusters, 4 clusters, 6 clusters, and 8 clusters. In all these trials, the performance was similar in that the clustering errors were smaller in GA-ADPCM than in ADPCM.

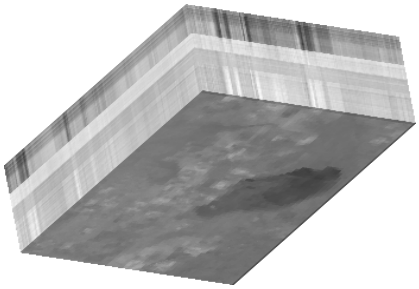


Figure 7 – The Lunar Crater Volcanic Field

Figure 8 – One image band of Jasper Ridge

First, a two-dimensional LMS (2D-LMS) was used in the ADPCM system. Figure 9 is one 64x64 block of the original image, and Figure 10 is the clustering result of the original image. In SOM, 4 clusters were used. Figure 11 is the clustering result using LMS only, and Figure 12 is the clustering result using GA-LMS. We can see that all of these clustering results are correlated, but small differences are apparent between the images. Our purpose is to reduce the differences, thus reducing clustering error. The total clustering error using GA-LMS (0.066406) is smaller than that using LMS (0.078125).

In GA, the individuals in each generation have different fitness scores. The fittest individual who has the smallest clustering error is saved. Figure 13 shows the fitness scores of different generations. In each generation, there is a lowest fitness score, a highest score, and an average score. The red circles represent the lowest fitness scores, the blue line represents the highest scores, and the green marks represent the average scores. We can see that the lowest fitness scores converge to a certain value. The coefficients corresponding to the individual for this value is optimal because they can minimize both the clustering error and the mean-square error.

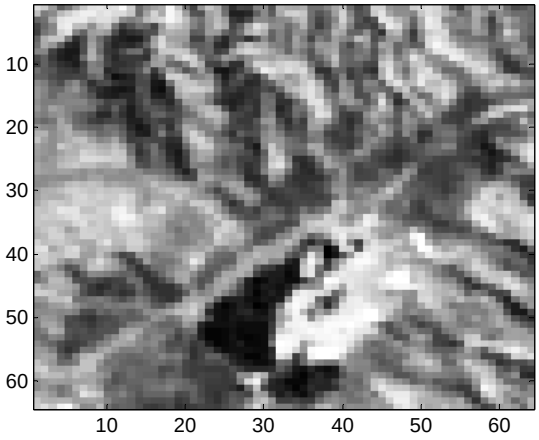


Figure 9 – One block of a image band of Jasper Ridge

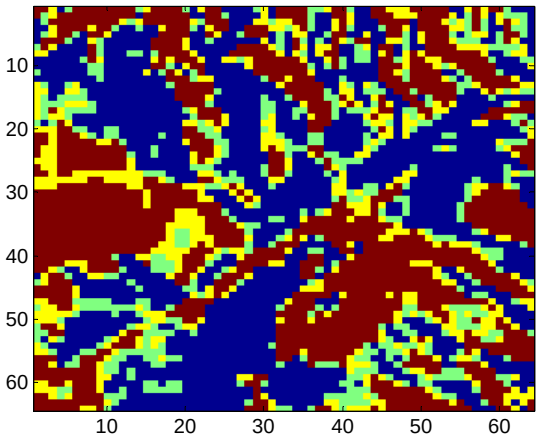


Figure 10 – Clustered image from original image

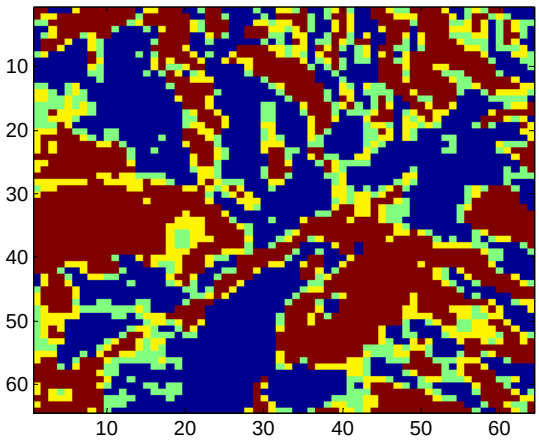


Figure 11 – Clustered image from LMS image

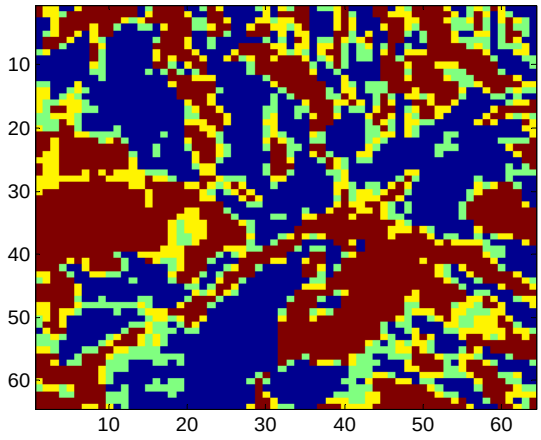


Figure 12 – Clustered image from GA-LMS image

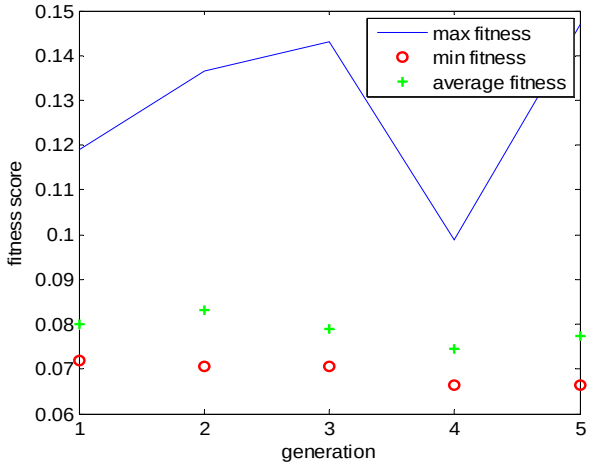


Figure 13 – Fitness score for GA-LMS

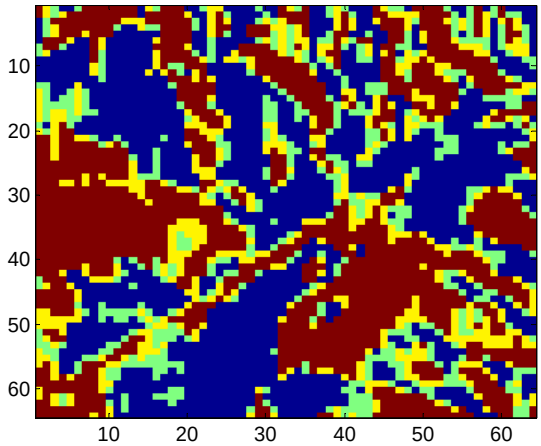


Figure 14 – Clustered image from EDS image

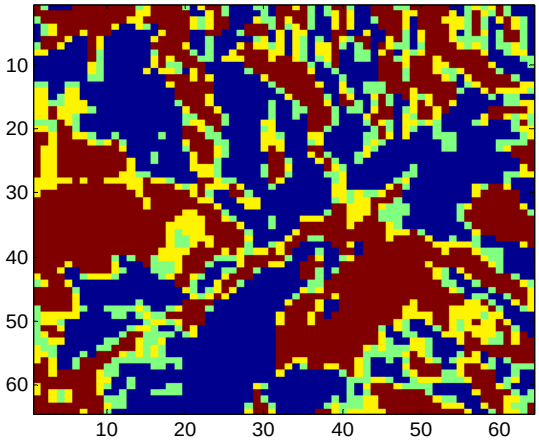


Figure 15 – Clustered image from GA-EDS image

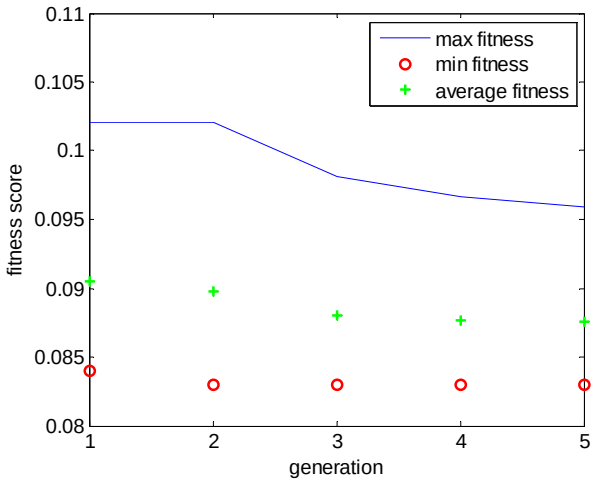


Figure 16 – Fitness score for GA-EDS

A 2D-EDS predictor was also tested in the ADPCM system. The same block of the original image was used. Figure 14 is the clustering result using only EDS. Figure 15 is the clustering result using GA-EDS. The total clustering error using GA-EDS is 0.083008, and the error using EDS is 0.091064. When EDS is used as the predictor, the performance is still improved by using GA-EDS. Figure 16 shows the fitness scores of different generations in GA-EDS.

Table 1 lists the clustering error comparison between LMS and GA-LMS and between EDS and GA-EDS. The block size is 64x64 pixels and the number of class is 4. The table gives the errors of blocks (1, 1) through (1, 8). As seen from the table, the GA-ADPCM performs better than the ADPCM in minimizing clustering error.

Table 1. Clustering Error Comparison between ADPCM and GA-ADPCM

Block Index	LMS	GA-LMS	EDS	GA-EDS
(1,1)	0.0424	0.03442	0.0527	0.04541
(1,2)	0.060303	0.049561	0.085938	0.070557
(1,3)	0.05127	0.040771	0.070801	0.061279
(1,4)	0.027588	0.023193	0.039551	0.032227
(1,5)	0.04248	0.027588	0.056641	0.04834
(1,6)	0.039063	0.037354	0.05127	0.050781
(1,7)	0.057129	0.049072	0.065186	0.05835
(1,8)	0.069336	0.065674	0.092773	0.080566

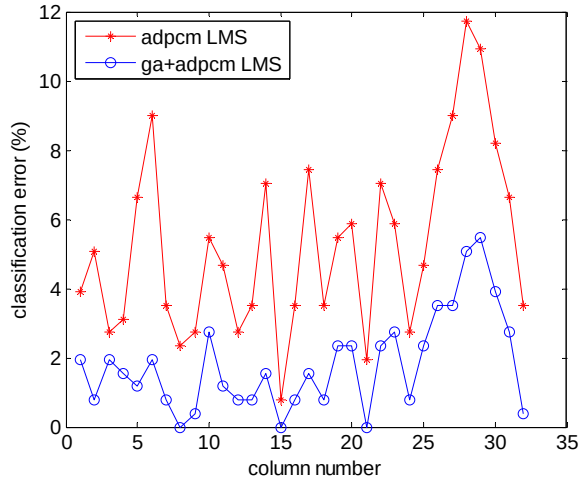


Figure 17 – Clustering error comparison between LMS and GA-LMS: block size=16, clusters=3

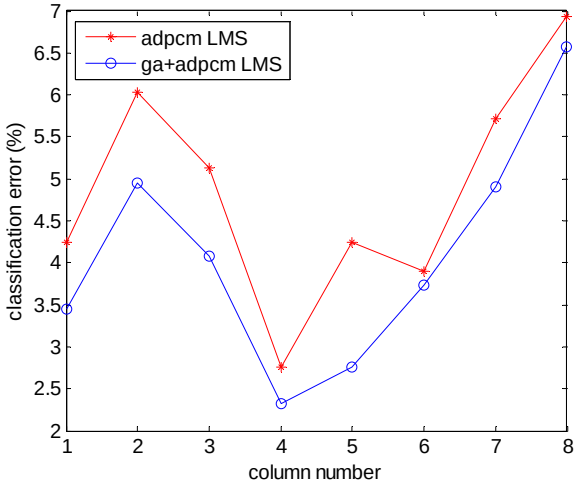


Figure 18 – Clustering error comparison between LMS and GA-LMS: block size=64, clusters=4

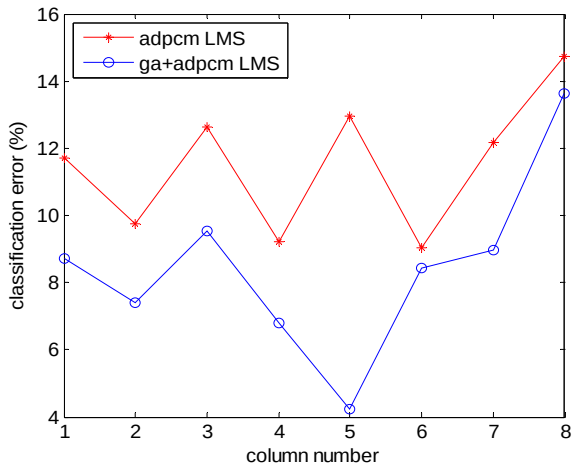


Figure 19 – Clustering error comparison between LMS and GA-LMS: block size=64, clusters=8

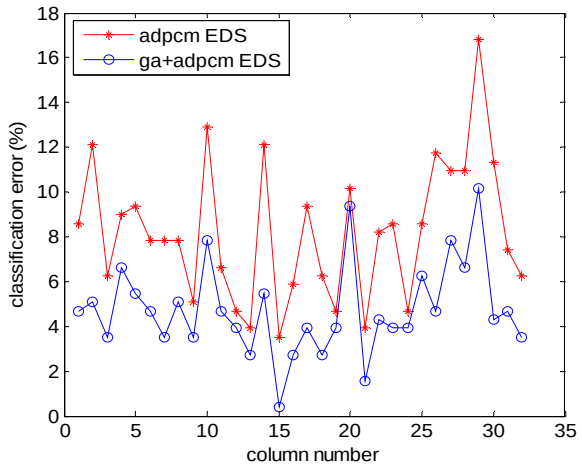


Figure 20 – Clustering error comparison between EDS and GA-EDS: block size=16, clusters=4

Different block sizes and cluster sizes were tested. Figures 17–20 illustrate the clustering-error comparison between the ADPCM and the GA-ADPCM. Red lines represent the error using the ADPCM, while blue lines are the error using the GA-ADPCM. In these figures, all the red lines are higher than the blue ones, meaning that the clustering errors are larger in ADPCM than in GA-ADPCM. Figure 18 and 19 also illustrate that the clustering errors are sensitive to the number of clusters. For the same block size, the number of clusters increases, the cluster error increases.

Feedback-Transform Performance

For this preliminary work, a subcube was taken from each of the LCVF and the JR images individually, which was used as the original image. A subcube was used to increase the speed of performance. Another important reason is that Matlab cannot handle the whole hyperspectral image. Six bands, bands 64 – 69, were selected arbitrarily from all AVIRIS bands, for this preliminary study. For each image band, the size is 128 x 128 pixels. Thus the subcube has size 128 x 128 pixels x 6 bands. Figure 21 shows one image band of the LCVF image subcube.

During the procedure, clustering was still using a SOM and the whole subcube was clustered into 6 clusters. Transform was performed by using 1-D DCT along the spectral domain. The image cube was reshaped into one-dimensional vector along spectral domain. The vector was cut into blocks and DCT and quantization were applied to it. After applying dequantization and inverse DCT, the data was reshaped back to an image cube, and then SOM clustering was applied to it.

According to the property of DCT, we set the first one-sixth of block size data to apply large-bit uniformly distributed quantization because these are the most significant data. For Method I, we use 8 bits. For Method II, we initially set it to 5 bits. For the rest of the data, a two-bit quantization was used. The clustering error threshold was set to be 3%. If the clustering error was larger than the threshold, we cut the block size into half for Method I or increased by 1 bit in the quantization of significant data for Method II. Next, the steps introduced in the Feed-back transform algorithm were applied again. Otherwise, we stopped.

Figure 22 shows the clustered image of the original LCVF. Table 2 shows the clustering errors using different block sizes for a settled number of bits, which is the result of applying Method I. To achieve an error smaller than 3%, the block size needs to be 24 or less. The DCT processing time in Matlab is also listed in Table 2. For larger block size, the less DCT processing time is needed. For smaller block size, the more processing time is needed. Figure 23 shows the clustered image after transform when using block size 24. Method I was also tested on the JR image. Figure 24 shows one image band of the JR image subcube. Figure 25 shows the clustered image of the original JR. Table 3 shows the clustering errors using different block sizes. To achieve the error threshold, the block size needs to be 48 or less. Figure 26 shows the clustered image after transform when using block size 48. Apparently, different images can get different suitable block sizes.

Method II was also applied to the JR cube. The DCT block size was fixed to be 48. Table 4 lists the clustering errors using different numbers of bits. It needs 8 bits to achieve the error threshold. This method requires fewer bits as compared to the 12 bits used to represent a pixel of AVIRIS data. The compression ratio for the whole subcube is also listed in Table 4 to represent the storage saving performance. For a higher compression ratio, the less storage is needed. For the whole uncompressed JR subcube, 1179648 bits are needed for the storage. After applying Method II and using 8 bits for the most significant data, 294912 bits are needed for the storage. The compression ratio is thereby 4:1.

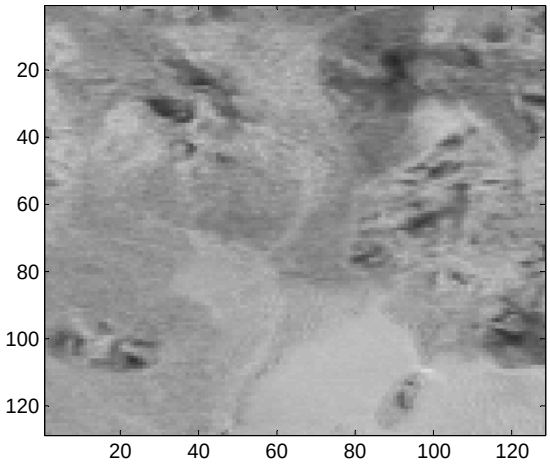


Figure 21 – One image band of the LCVF subcube

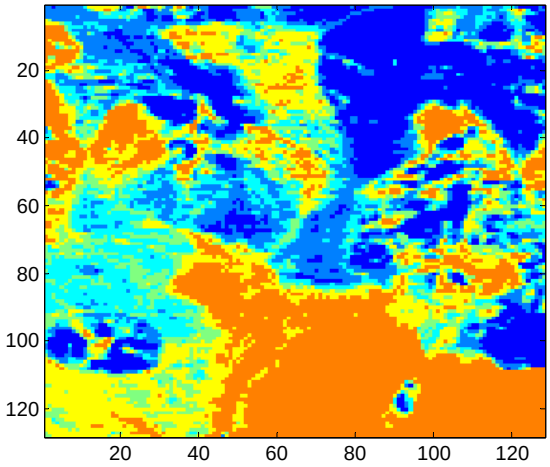


Figure 22 – Clustered image of original image – 6 clusters

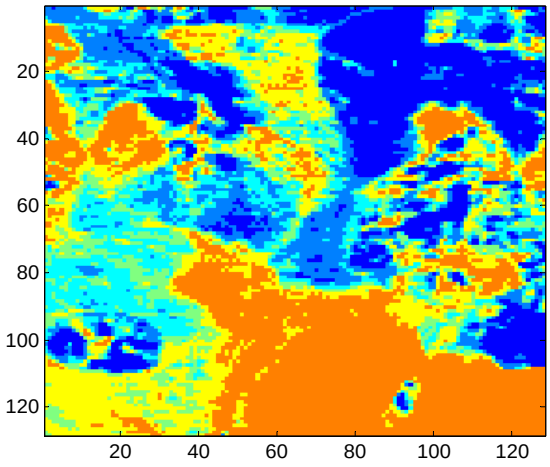


Figure 23 – Clustered image after transform – 6 clusters

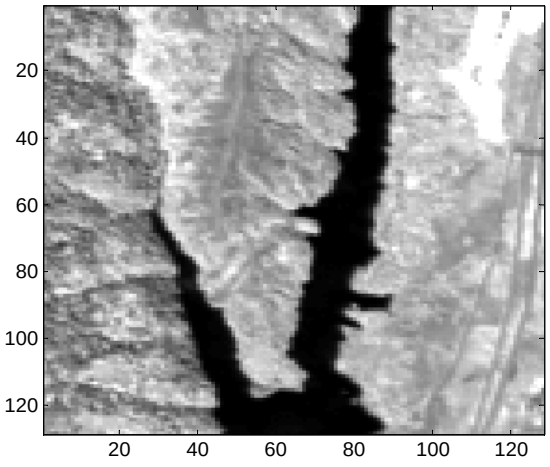


Figure 24 – One image band of the JR subcube

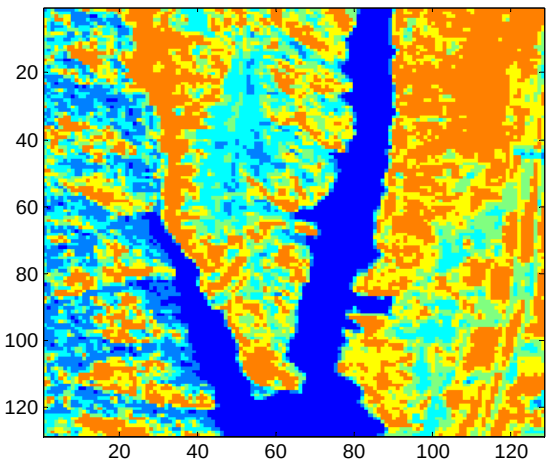


Figure 25 – Clustered image of original image – 6 clusters

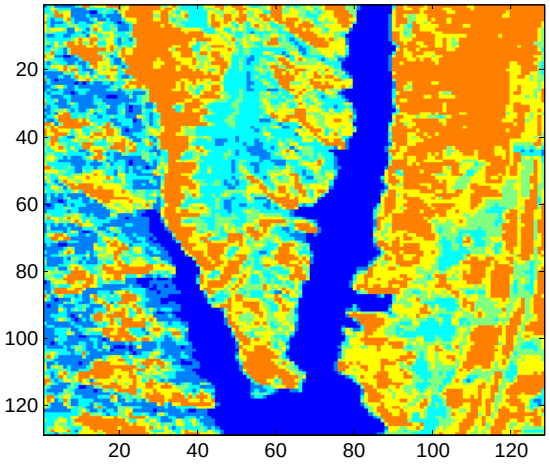


Figure 26 – Clustered image after transform – 6 clusters

Figure 27 illustrates the mean spectral signatures of the clusters for both original image and the image after applying transform compression and decompression. This result is based on the JR image and applying Method I for transform compression. The "before" and "after" spectral signatures are very similar, which shows that the compression preserved the cluster distinctions.

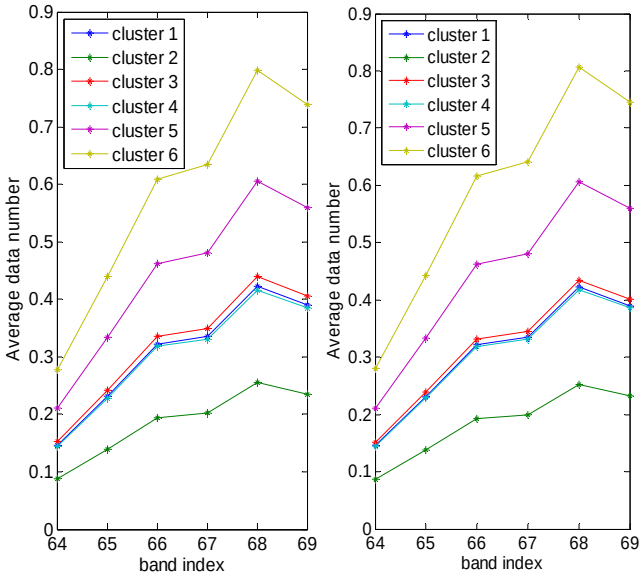


Figure 27 – Mean spectral signatures of the SOM clusters identified in the Jasper Ridge image. Left: from the original image. Right: from the image after applying DCT compression and decompression

Table 2. Clustering Errors using Different Block Sizes in LCVF

Iteration	1	2	3	4	5	6
Block size	768	384	192	96	48	24
Cluster error	0.149	0.109	0.078	0.050	0.036	0.029
DCT processing time (s)	0.427	0.435	0.527	0.742	1.097	1.890

Table 3. Clustering Errors using Different Block Sizes in JR

Iteration	1	2	3	4	5
Block size	768	384	192	96	48
Cluster error	0.0980	0.0728	0.0469	0.0317	0.0218
DCT processing time (s)	0.4246	0.4408	0.5379	0.7165	1.1376

Table 4. Clustering Errors using Different Number of Bits in JR

Iteration	1	2	3	4
Bit	5	6	7	8
Cluster error	0.1292	0.0657	0.0367	0.0223
Compression ratio	4.8:1	4.5:1	4.2:1	4:1

5. CONCLUSION

A new structure combining image compression and image clustering schemes was investigated in this paper. Two corresponding schemes were modeled. One was the ADPCM scheme, and the other the transform scheme. The corresponding algorithms, GA-ADPCM and Feedback-transform were simulated, respectively. For the ADPCM scheme, the filter coefficients in ADPCM were optimized by using the genetic algorithm and the LMS and EDS predictors were used. In the genetic algorithm, the fitness function used the clustering error between pre- and post-compression images so that the filter coefficients were also changed to optimize the clustering error. The simulation results demonstrate that the algorithm has good performance. However, the cost is the computation complexity. For the transform scheme, two methods were tested. With fixed number of bits in quantization, the suitable block size for DCT was chosen. With fixed block size, the suitable number of bits was chosen. For different hyperspectral data, the suitable block size and number of bits were different. This scheme aims to optimize computation time and storage. In the future, more sophisticated classification methods, such as SOM-based hybrid ANN [9], will be applied. For the ADPCM scheme, a multiplier-free adaptive filter [10] combined with the genetic algorithm will also be investigated. For the transform scheme, different types of image transforms will be tested.

REFERENCES

[1] G. Shaw and D. Manolakis, "Signal Processing for Hyperspectral Image Exploitation," Signal Processing Magazine, Vo1.19, No.1, January 2002.

[2] E. Merényi, W. H. Farrand, and P. Tracadas, "Mapping Surface Materials on Mars From Mars Pathfinder Spectral Images With HYPEREYE," Proc. International Conference on Information Technology (ITCC 2004), vol. II, pp. 607 - 614. April 5-7, 2004.

[3] G. F. Xu, T. Bose, and J. Schroeder, "The Euclidean direction search algorithm for adaptive filtering," Circuits and Systems, 1999. ISCAS '99. Proceedings of the 1999 IEEE International Symposium on, vol.3, pp.146-149 vol.3, Jul 1999.

[4] T. Bose, Digital Signal and Image Processing, John Wiley, 2004.

[5] J.D. Griesbach and D.M. Etter, "Fitness-based exponential probabilities for genetic algorithmsapplied to adaptive IIR filtering," Asilomar Conference of Signal, Systems, and Computers, vol. 1, pp. 523- 527, 1998.

[6] Web site http://en.wikipedia.org/wiki/Genetic_algorithms.

[7] S. Ovaska, Computationally Intelligent Hybrid Systems, John Wiley, 2005.

[8] Web site http://aviris.jpl.nasa.gov/html/data.html

[9] T. Villmann, E. Merényi, and B. Hammer, "Neural Maps in Remote Sensing Image Analysis," Neural Networks, 16: pp. 389 – 403, 2003.

[10] T. Bose, A. Venkatachalam, and R. Thamvichai, "Multiplierless adaptive filtering," Digital Signal Processing, vol. 12, pp. 107-118, 2002.

BIOGRAPHY

Bei Xie *received the M.S. degree in electrical engineering from East China Normal University, Shanghai, China, in 2005. She is now a Ph.D. student in Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, U.S.A.*

Tamal Bose *received the Ph.D. degree in electrical engineering from Southern Illinois University in 1988. After faculty positions at the Citadel and the University of Colorado, he joined Utah State University in 2000, where served as the Department Head and Professor of Electrical and Computer Engineering from 2003-2007. Currently, he is Professor of the Bradley Department of Electrical and Computer Engineering at Virginia Tech. He is also the Associate Director of Wireless@VT.*

The research interests of Dr. Bose include adaptive filtering algorithms, nonlinear effects in digital filters, and multidimensional system theory. He is author of the text Digital Signal and Image Processing, John Wiley, 2004. He is also the author or co-author of over 100 technical papers. Dr. Bose served as the Associate Editor for the IEEE Transactions on Signal Processing from 1992 to 1996. He is currently on the editorial board of the IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Japan. He also served on the organizing committees of several international conferences. Dr. Bose received the 2002 Researcher of the Year award from the College of Engineering at Utah State University and the 2002 Teacher of the Year award from the department of Electrical and Computer Engineering. He received the Researcher of the Year and Service Person of the Year awards at the University of Colorado at Denver. He also received two Exemplary Researcher awards from the Colorado Advanced Software Institute. He is an IEEE EAC program evaluator and a member of the DSP Technical Committee for the IEEE Circuits and Systems society.

Erzsébet Merényi *received her M.Sc. in mathematics (1975) and Ph.D. in computer science (1980) at Szeged (Attila József) University, Hungary. She is currently a research professor in the Electrical and Computer Engineering Department of Rice University, Houston, Texas, U.S. A. She previously worked as a staff scientist at the Lunar and Planetary Laboratory of the University of Arizona, Tucson, Arizona. Her interests include artificial neural networks, self-organized learning, manifold learning, segmentation and classification of high-dimensional patterns, data fusion, data mining, knowledge discovery, and application to information extraction from multi- and hyperspectral remote sensing imagery from Earth and space science missions, from hyperspectral medical imagery, and from multi-variate medical data.*